

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	25390429	@ad<"20040112"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:16
L2	10611	trace near5 generat\$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:17
L3	450	trace adj cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:18
L4	715513	(entry entries)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:18
L5	26	1 and 2 and 3 with 4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:32
L6	33197	(range threshold) near5 address\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:33
L7	1	5 and 6	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:33

EAST Search History

L8	124034	(contiguous\$2 range set) near5 address\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:35
L9	7	5 and 8	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:34
L10	52962	(contiguous\$2 range threshold limit\$3) near5 address\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:57
L11	4	5 and 10	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:38
L12	762	trace near3 cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:38
L13	8	1 and 2 and 12 with 4 and 10	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:58
L14	495	711/217.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:49

EAST Search History

L15	275	711/218.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:53
L16	580	711/200.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:54
L17	4027	365/230.03.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:54
L18	2061	365/230.01.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:54
L19	54438	(contiguous\$2 range threshold limit\$3 bound) near5 address\$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 12:57
L20	8	1 and 2 and 12 with 4 and 19	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:03
L21	714	711/3.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:21

EAST Search History

L22	403	712/208.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:21
L23	152	712/211.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:23
L24	0	1 and (14 15 16 17 18 21 22 23) and (5 9 11 13 20)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:27
L25	30	1 and (14 15 16 17 18 21 22 23) and (3 12)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:30
L26	8	1 and (14 15 16 17 18 21 22 23) and (3 12) and 19	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:33
L27	2	smaus.in.	US-PGPUB	OR	ON	2007/02/02 13:33
L28	2	27 and "trace generator".clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/02/02 13:34



Welcome United States Patent and Trademark Office

[Search Session History](#)[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Fri, 2 Feb 2007, 2:11:55 PM EST

Edit an existing query or
compose a new query in the
Search Query Display.

Search Query Display

Select a search number (#)
to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries

- #1 ((trace cache<in>metadata) <and> (entries<in>metadata))
<and> (restrict<in>metadata)
- #2 ((trace<in>metadata) <and> (entries<in>metadata))<and>
(range<in>metadata)
- #3 ((trace cache<in>metadata) <and>
(contiguous<in>metadata))<and> (addresses<in>metadata)
- #4 ((trace generator<in>metadata) <and>
(contiguous<in>metadata))<and> (addresses<in>metadata)
- #5 ((trace generator<in>metadata) <and>
(range<in>metadata))<and> (addresses<in>metadata)
- #6 ((trace<in>metadata) <and> (range<in>metadata))<and>
(address<in>metadata)
- #7 ((trace cache<in>metadata) <and> (range<in>metadata))
<and> (address<in>metadata)
- #8 ((trace generator<in>metadata) <and>
(range<in>metadata))<and> (address<in>metadata)
- #9 ((trace<in>metadata) <and> (restrict<in>metadata))<and>
(range<in>metadata)
- #10 ((trace<in>metadata) <and> (range<in>metadata))<and>
(predetermined<in>metadata)

Indexed by
 Inspect

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

Full text available:  [pdf\(3.08 MB\)](#)[terms](#)

In this paper a new method to improve the utilization of main memory systems is presented. The new method is based on prestoring in main memory a number of query answers, each evaluated out of a single memory page. To this end, the ideas of page-answers and page-traces are formally described and their properties analyzed. The query model used here allows for selection, projection, join, recursive queries as well as arbitrary combinations. We also show how to apply the approach under update ...

Keywords: artificial intelligence, databases, page access

4 [Application-specific optimizations: Two-level mapping based cache index selection for packet forwarding engines](#)

Kaushik Rajan, R. Govindarajan

September 2006 **Proceedings of the 15th international conference on Parallel architectures and compilation techniques PACT '06**

Publisher: ACM Press

Full text available:  [pdf\(802.85 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Packet forwarding is a memory-intensive application requiring multiple accesses through a trie structure. The efficiency of a cache for this application critically depends on the placement function to reduce conflict misses. Traditional placement functions use a one-level mapping that naively partitions trie-nodes into cache sets. However, as a significant percentage of trie nodes are not useful, these schemes suffer from a non-uniform distribution of useful nodes to sets. This in turn results i ...

Keywords: cache architectures, network processors

5 [Exploiting perception in high-fidelity virtual environments: Exploiting perception in high-fidelity virtual environments](#)

Additional presentations from the 24th course are available on the citation page

Mashhuda Glencross, Alan G. Chalmers, Ming C. Lin, Miguel A. Otaduy, Diego Gutierrez
July 2006 **ACM SIGGRAPH 2006 Courses SIGGRAPH '06**

Publisher: ACM Press

Full text available:  [pdf\(5.07 MB\)](#)  [mov\(68:6 MIN\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The objective of this course is to provide an introduction to the issues that must be considered when building high-fidelity 3D engaging shared virtual environments. The principles of human perception guide important development of algorithms and techniques in collaboration, graphical, auditory, and haptic rendering. We aim to show how human perception is exploited to achieve realism in high fidelity environments within the constraints of available finite computational resources. In this course w ...

Keywords: collaborative environments, haptics, high-fidelity rendering, human-computer interaction, multi-user, networked applications, perception, virtual reality

6 [Parallel execution of prolog programs: a survey](#)

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo
July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 23 Issue 4

Publisher: ACM Press

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Full text available:  pdf(1.95 MB)[terms](#)

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

7 [Cache coherence in large-scale shared-memory multiprocessors: issues and comparisons](#)



David J. Lilja

September 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 3**Publisher:** ACM PressFull text available:  pdf(3.12 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 [A proposal for a new hardware cache monitoring architecture](#)



Martin Schulz, Jie Tao, Jürgen Jeitner, Wolfgang Karl

June 2002 **ACM SIGPLAN Notices , Proceedings of the 2002 workshop on Memory system performance MSP '02**, Volume 38 Issue 2 supplement**Publisher:** ACM PressFull text available:  pdf(1.23 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The analysis of the memory access behavior of applications, an essential step for a successful cache optimization, is a complex task. It needs to be supported with appropriate tools and monitoring facilities. Currently, however, users can only rely on either simulation based approaches, which deliver a large degree of detail but are restricted in their applicability, or on hardware counters embedded into processors, which allow to keep track of very few, mostly global events and hence only provi ...

9 [LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance](#)



Song Jiang, Xiaodong Zhang

June 2002 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '02**, Volume 30 Issue 1**Publisher:** ACM PressFull text available:  pdf(290.24 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)


Although LRU replacement policy has been commonly used in the buffer cache management, it is well known for its inability to cope with access patterns with weak locality. Previous work, such as LRU-K and 2Q, attempts to enhance LRU capacity by making use of additional history information of previous block references other than only the recency information used in LRU. These algorithms greatly increase complexity and/or can not consistently provide performance improvement. Many recently proposed ...

10 [Dynamic hot data stream prefetching for general-purpose programs](#)



Trishul M. Chilimbi, Martin Hirzel

May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation PLDI '02**, Volume 37 Issue 5

Publisher: ACM PressFull text available:  pdf(210.85 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Prefetching data ahead of use has the potential to tolerate the growing processor-memory performance gap by overlapping long latency memory accesses with useful computation. While sophisticated prefetching techniques have been automated for limited domains, such as scientific codes that access dense arrays in loop nests, a similar level of success has eluded general-purpose programs, especially pointer-chasing codes written in languages such as C and C++. We address this problem by describing ...

Keywords: data reference profiling, dynamic optimization, dynamic profiling, memory performance optimization, prefetching, temporal profiling

11 [A predicate-based caching scheme for client-server database architectures](#)


Arthur M. Keller, Julie Basu

January 1996 **The VLDB Journal — The International Journal on Very Large Data****Bases**, Volume 5 Issue 1**Publisher:** Springer-Verlag New York, Inc.Full text available:  pdf(162.80 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

We propose a new client-side data-caching scheme for relational databases with a central server and multiple clients. Data are loaded into each client cache based on queries executed on the central database at the server. These queries are used to form predicates that describe the cache contents. A subsequent query at the client may be satisfied in its local cache if we can determine that the query result is entirely contained in the cache. This issue is called *cache completeness*. A separ ...

Keywords: Cache completeness, Cache currency, Caching, Multiple clients, Relational databases

12 [A parallel, incremental, mostly concurrent garbage collector for servers](#)

 Katherine Barabash, Ori Ben-Yitzhak, Irit Gof, Elliot K. Kolodner, Victor Leikehman, Yoav Ossia, Avi Owshanko, Erez Petrank
November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6**Publisher:** ACM PressFull text available:  pdf(683.50 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Multithreaded applications with multigigabyte heaps running on modern servers provide new challenges for garbage collection (GC). The challenges for "server-oriented" GC include: ensuring short pause times on a multigigabyte heap while minimizing throughput penalty, good scaling on multiprocessor hardware, and keeping the number of expensive multicycle fence instructions required by weak ordering to a minimum. We designed and implemented a collector facing these demands building on th ...

Keywords: Garbage collection, JVM, concurrent garbage collection

13 [Collision detection and proximity queries](#)

 Sunil Hadap, Dave Eberle, Pascal Volino, Ming C. Lin, Stephane Redon, Christer Ericson
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04****Publisher:** ACM PressFull text available:  pdf(11.22 MB) Additional Information: [full citation](#), [abstract](#)

This course will primarily cover widely accepted and proved methodologies in collision detection. In addition more advanced or recent topics such as continuous collision detection, ADFs, and using graphics hardware will be introduced. When appropriate the methods discussed will be tied to familiar applications such as rigid body and cloth simulation, and will be compared. The course is a good overview for those developing applications in physically based modeling, VR, haptics, and robotics.

14 Special issue: AI in engineering



D. Sriram, R. Joobhani

April 1985 **ACM SIGART Bulletin**, Issue 92

Publisher: ACM Press

Full text available: pdf(8.79 MB) Additional Information: [full citation](#), [abstract](#)

The papers in this special issue were compiled from responses to the announcement in the July 1984 issue of the SIGART newsletter and notices posted over the ARPAnet. The interest being shown in this area is reflected in the sixty papers received from over six countries. About half the papers were received over the computer network.

15 Introduction to real-time ray tracing: The RTRT core



Ingo Wald

July 2005 **ACM SIGGRAPH 2005 Courses SIGGRAPH '05**

Publisher: ACM Press

Full text available: pdf(1.13 MB) Additional Information: [full citation](#), [abstract](#), [references](#)

The overall design decisions of the RTRT/OpenRT framework are described in detail in [Wald04]. To summarize the most important points, we have chosen to only support triangles, to exploit SIMD extensions in a data-parallel way, to optimize for memory and caches, and to use BSP trees as an acceleration structure. In this chapter, we are now going to discuss the actual algorithms and implementation of these topics in more detail.

16 Data cache management using frequency-based replacement



John T. Robinson, Murthy V. Devarakonda

April 1990 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '90**, Volume 18 Issue 1

Publisher: ACM Press

Full text available: pdf(991.05 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We propose a new frequency-based replacement algorithm for managing caches used for disk blocks by a file system, database management system, or disk control unit, which we refer to here as data caches. Previously, LRU replacement has usually been used for such caches. We describe a replacement algorithm based on the concept of maintaining reference counts in which locality has been "factored out". In this algorithm replacement choices are made using a combination of reference f ...

17 Executing compressed programs on an embedded RISC architecture



Andrew Wolfe, Alex Chanin

December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture MICRO 25**, Volume 23 Issue 1-2

Publisher: IEEE Computer Society Press, ACM Press

Full text available: pdf(1.53 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

18 A case for dynamic view management


 Yannis Kotidis, Nick Roussopoulos
 December 2001 **ACM Transactions on Database Systems (TODS)**, Volume 26 Issue 4
 Publisher: ACM Press

Full text available:  pdf(892.57 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Materialized aggregate views represent a set of redundant entities in a data warehouse that are frequently used to accelerate On-Line Analytical Processing (OLAP). Due to the complex structure of the data warehouse and the different profiles of the users who submit queries, there is need for tools that will automate and ease the view selection and management processes. In this article we present DynaMat, a system that manages dynamic collections of materialized aggregate views in a data warehouse ...

Keywords: Data cube, OLAP, data warehousing, materialized views

19 [The automatic improvement of locality in storage systems](#)

 Windsor W. Hsu, Alan Jay Smith, Honesty C. Young
 November 2005 **ACM Transactions on Computer Systems (TOCS)**, Volume 23 Issue 4
 Publisher: ACM Press


Full text available:  pdf(2.58 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Disk I/O is increasingly the performance bottleneck in computer systems despite rapidly increasing disk data transfer rates. In this article, we propose Automatic Locality-Improving Storage (ALIS), an introspective storage system that automatically reorganizes selected disk blocks based on the dynamic reference stream to increase effective storage performance. ALIS is based on the observations that sequential data fetch is far more efficient than random access, that improving seek distances prod ...

Keywords: Data layout optimization, block layout, data reorganization, data restructuring, defragmentation, disk technology trends, locality improvement, prefetching

20 [I/O reference behavior of production database workloads and the TPC benchmarks— an analysis at the logical level](#)

 Windsor W. Hsu, Alan Jay Smith, Honesty C. Young
 March 2001 **ACM Transactions on Database Systems (TODS)**, Volume 26 Issue 1
 Publisher: ACM Press

Full text available:  pdf(5.42 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As improvements in processor performance continue to far outpace improvements in storage performance, I/O is increasingly the bottleneck in computer systems, especially in large database systems that manage huge amounts of data. The key to achieving good I/O performance is to thoroughly understand its characteristics. In this article we present a comprehensive analysis of the logical I/O reference behavior of the peak production database workloads from ten of the world's largest corporations ...

Keywords: I/O, TPC benchmarks, caching, locality, prefetching, production database workloads, reference behavior, sequentiality, workload characterization

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)